Notes

Inverse Depth Parametrization for Monocular SLAM

Sally Hui

These are a work in progress!

1.1 Introduction

Delayed and Undelayed Initialization

Delayed initialization means adding features only when their depth uncertainty is sufficiently low. This means that they cannot be used in the map for pose estimation until they are finally added. This also means that features that are far away or close to the motion epipole may not make it into the map, as they are rejected before inclusion into the map. Previous work using this approach has involved: waiting till the measurement equation appears Gaussian; and, using a 3D semi-infinite ray in the main map to represent everything except the depth of a feature and using a particle filter to refine the estimate, but **is not added to the state vector until included** (I think this is a good point, that you can add points to the map but not include them in the state).

Undelayed initialization means new features can immediately be used to refine the motion estimate; while highly uncertain depth provides little information on translational movement, they are still very good as bearing references for rotational movement.

Points at Infinity

Points at infinity exhibit no parallax due to extreme depth. Similar to above, they cannot be used to estimate camera translation, but are perfect for estimating rotation, as they are great bearings. When all features are infinite (e.g. very-large-scale outdoor scenes), SLAM turns the camera into a visual compass.

We can treat features as all starting with infinite depth, and "coming in" when sufficient parallax is observed. An appropriate scheme must be chosen so that finite depth can be proven given sufficient motion, and uncertainty in depth can be represented for even features with infinite depth. The reason here is that, even if we observe no parallax for a feature, that gives us a lower bound on its depth.

Inverse Depth Representation

There is clearly an advantage to undelayed representation. Explicit parametrization of the *inverse depth* of a feature along a semiinfinite ray from the position it was first viewed allows a Gaussian distribution to cover uncertainty in depth to nearby to infinity. Thus, it can be inserted into the map right at the start.

We use a linearity index to indicate how well low and high parallax features can be represented through inverse depth parametrization.

Inverse depth representation requires six parameters, rather than the 3 of XYZ coding. The linearity index can indicate when a feature can be safely switched back to XYZ encoding. This means that features can

adopt a sort of hybrid scheme, where points that are low parallax are represented with the inverse depth parametrization, while points that are high parallax are represented with XYZ encoding.

The projective nature of a camera means that the image measurement process is nearly linear in the inverse depth coordinate. The proposed inverse depth parametrization is *relative to the position from which features were first observed*, which preserves this linearity and means the Gaussian distribution is uniquely well behaved.

1.2 State Vector

Camera Motion

A constant angular and linear velocity model is used to model handheld camera motion. The camera state x_v is composed of pose terms: r_{WC} camera optical center position and q_{WC} quaternion defining orientation, and linear and angular velocity v^W and ω^C relative to world frame W and camera frame C, respectively.

The linear and angular accelerations a^W and α^C produce impulses of linear and angular velocity:

$$\begin{aligned} V^W &= a^W \Delta t \\ \Omega^C &= \alpha^C \Delta t \end{aligned} \tag{1.1}$$

The state update equation for the camera is then:

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r}^{WC}_{k+1} \\ \mathbf{q}^{WC}_{k+1} \\ \mathbf{v}^{W}_{k+1} \\ \omega^{C}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r}^{WC}_{k} + \left(\mathbf{v}^{W}_{k} + \mathbf{V}^{W}_{k}\right) \Delta t \\ \mathbf{q}^{WC}_{k} \times \mathbf{q}\left(\left(\omega^{C}_{k} + \Omega^{C}\right) \Delta t\right) \\ \mathbf{v}^{W}_{k} + \mathbf{V}^{W} \\ \omega^{C}_{k} + \Omega^{C} \end{pmatrix} \tag{1.2}$$

The first row is simply adding the current velocity to the impulse to get an updated velocity, and using the time delta to get the change in position.

The second row is an update for the quaternion representing the rotation of the camera; $\mathbf{q}\left(\left(\omega^C_k + \Omega^C\right) \Delta t\right)$ is the quaternion representing the rotation from $\left(\omega^C_k + \Omega^C\right) \Delta t$

The two bottom rows are simply adding the current velocity to the calculated impulse.

Euclidean XYZ Point Parametrization

As standard, the i^{th} point is

$$\mathbf{x}_i = \begin{pmatrix} X_i & Y_i & Z_i \end{pmatrix}^T . \tag{1.3}$$

Inverse Depth Point Parametrization

A 3D point is defined by the 6-D state vector:

$$\mathbf{y}_i = \begin{pmatrix} x_i & y_i & z_i & \theta_i & \phi_i & \rho_i \end{pmatrix}^T . \tag{1.4}$$

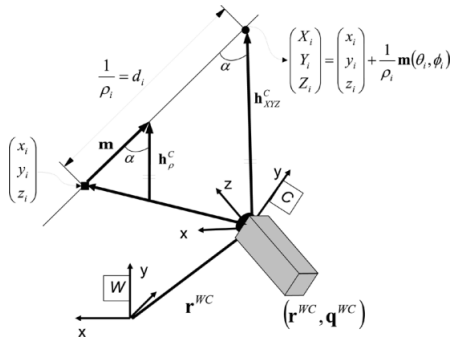

Figure 1.1: Illustration of parameters, using the inverse depth scheme

This models a 3D point located at

$$\mathbf{x}_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right), \tag{1.5}$$

where

$$\mathbf{m} = \left(\cos\,\phi_i \sin\,\theta_i, -\sin\,\phi_i, \cos\,\phi_i \cos\,\theta_i\right)^T. \tag{1.6}$$

x_i, y_i, and z_i are the camera optical center from which the point was first observed, and θ_i and ϕ_i are azimuth and elevation respectively (coded in the world frame), which defines the unit vector $m(\theta_i, \phi_i)$.

The depth along the ray d_i is encoded by its inverse $\rho_i = 1/d_i$.

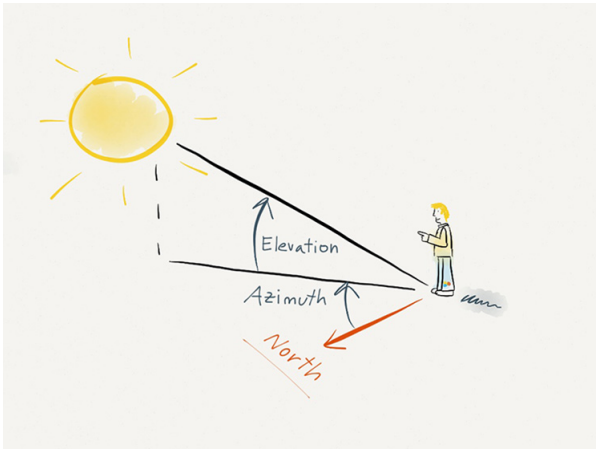

Figure 1.2: Illustration of elevation and azimuth

Full State Vector

The whole state vector contains the camera and all of the map features,

$$\mathbf{x} = \left(\mathbf{x}_v^\top, \mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_n^\top\right)^\top. \tag{1.7}$$

1.3 Measurement Equation

A point $y_i(x_i)$ defines a ray coded by a direction vector in the camera frame $\mathbf{h}^C = (h_x, h_y, h_z)^\top$.

To represent a point in XYZ coordinates in the camera frame, simply apply the transform that takes points from the world frame to camera frame. This can also be stated as subtracting the position of the camera and applying the inverse rotation that defines the pose of the camera:

$$\mathbf{h}^C = \mathbf{h}^C_{XYZ} = \mathbf{R}^{CW} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} - \mathbf{r}^{WC}. \tag{1.8}$$

To represent a point inverse depth,

$$\mathbf{h}^C = \mathbf{h}^C_\rho = \mathbf{R}^{CW} \left(\rho_i \left(\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} - \mathbf{r}^{WC} \right) + \mathbf{m}\left(\theta_i, \phi_i\right) \right) \tag{1.9}$$

Compare it with Equation 1.5, which has been normalized with the inverse depth. It is normalized so that we don't divide by ρ_i ($\rho_i = 0$ is safe).

The camera observes the projection of each point using the pinhole model, which gives image coordinates of

$$\mathbf{h} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - \dfrac{f}{d_x}\dfrac{h_x}{h_z} \\ v_0 - \dfrac{f}{d_y}\dfrac{h_y}{h_z} \end{pmatrix} \tag{1.10}$$

Where f is focal length, and d_x and d_y are the pixel size. This is the same as our normal pinhole projection model, our $f_x = f/d_x$ and $f_y = f/d_y$.

At low parallax (that is, it is far away, d_i is large and ρ_i is small), Equation 1.9 reduces to

$$\mathbf{h}^C \approx \mathbf{R}^{CW} \left(\mathbf{m}\left(\theta_i, \phi_i\right) \right) \tag{1.11}$$

1.4 Measurement Equation Linearity

The more linear a measurement equation is, the better a Kalman filter based on it performs.

Linearized Propagation of a Gaussian

We want to measure how linear a measurement equation is when evaluated around a value of a Gaussian variable. This is analagous to measuring the curvature of a multivariate function; indeed, the final expression mirrors the common expression for the curvature of a function parameterized by time. The difference is that we need to account for the Gaussian distribution of the variable, effectively by normalizing with the standard deviation.

Let x be an uncertain variable with a Gaussian distribution, $x \sim N(\mu_x, \sigma_x^2)$. Transforming x by f,

$$y \sim N\left(\mu_y, \sigma_y^2\right),\ \mu_y = f\left(\mu_x\right),\ \sigma_y^2 = \left.\frac{\partial f}{\partial x}\right|_{\mu_x} \sigma_x^2 \left.\frac{\partial f}{\partial x}\right|_{\mu_x}^\top \tag{1.12}$$

The taylor expansion for the first derivative gives

$$\frac{\partial f}{\partial x}(\mu_x + \Delta x) \approx \left.\frac{\partial f}{\partial x}\right|_{\mu_x} + \left.\frac{\partial^2 f}{\partial x^2}\right|_{\mu_x} \Delta x. \tag{1.13}$$

Comparing the value of the derivative at the interval center μ_x with the value at the extremes $\mu_x \pm 2\sigma_x$ using the linearity index,

$$L = \left| \frac{\left.\frac{\partial^2 f}{\partial x^2}\right|_{\mu_x} 2\sigma_x}{\left.\frac{\partial f}{\partial x}\right|_{\mu_x}} \right|. \tag{1.14}$$

Although this is called a "*linearity index*", $L \approx 0$ **indicates that the function is linear in the interval, and Gaussianity is preserved during the transformation.**

Linearity of XYZ Parametrization

A point's location error d is encoded as Gaussian in depth,

$$D = d_0 + d, \qquad d \sim N\left(0, \sigma_d^2\right). \tag{1.15}$$

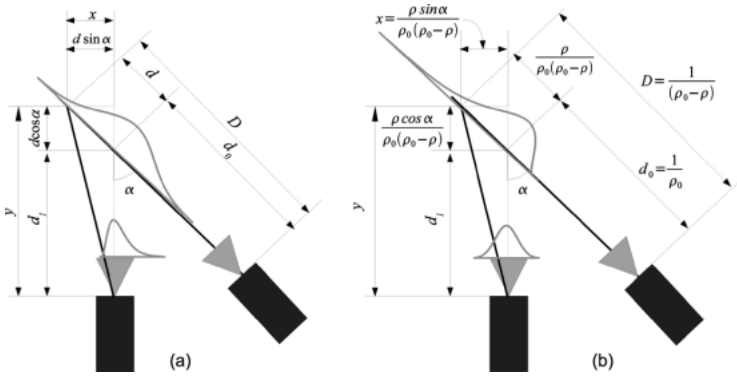

Figure 1.3: Depth uncertainty illustration

Accounting for this error d, the first coordinate of the 2D image of the point in the second camera is

$$u = \frac{x}{y} = \frac{d \sin\,\alpha}{d_1 + d\cos\,\alpha} \tag{1.16}$$

Applying the same linearity index expression from 1.14 gives

$$L_d = \left| \frac{(\partial^2 u/\partial d^2)2\sigma_d}{\partial u/\partial d} \right| = \frac{4\sigma_d}{d_1} \left|\cos\,\alpha\right| \tag{1.17}$$

Linearity of Inverse Depth Parametrization

Similarly, but with the depth error as Gaussian in inverse depth,

$$D = \frac{1}{\rho_0 - \rho}, \qquad \rho \sim N\left(0, \sigma_\rho^2\right) \tag{1.18}$$

$$d = D - d_0 = \frac{\rho}{\rho_0\left(\rho_0 - \rho\right)} \qquad d_0 = \frac{1}{\rho_0} \tag{1.19}$$

Giving

$$u = \frac{x}{y} = \frac{d \sin\,\alpha}{d_1 + d\cos\,\alpha} = \frac{\rho \sin\,\alpha}{\rho_0 d_1\left(\rho_0 - \rho\right) + \rho\cos\,\alpha} \tag{1.20}$$

and linearity index

$$L_\rho = \left|\frac{(\partial^2 u/\partial\rho^2)2\sigma_\rho}{\partial u/\partial\rho}\right| = \frac{4\sigma_\rho}{\rho_0}\left|1 - \frac{d_0}{d_1}\cos\alpha\right|. \tag{1.21}$$

1.5 Feature Initialization

We assign a general Gaussian prior in inverse depth to encode that the point has to be in front of the camera.

The initial location for a newly initialized feature is

$$\hat{\mathbf{y}}\left(\hat{\mathbf{r}}^{WC}, \hat{\mathbf{q}}^{WC}, \mathbf{h}, \rho_0\right) = \begin{pmatrix}\hat{x}_i & \hat{y}_i & \hat{z}_i & \hat{\theta}_i & \hat{\phi}_i & \hat{\rho}_i\end{pmatrix}^\top \tag{1.22}$$

Whcih is a function of the current pose estimate $\hat{r}^{WC}$, $\hat{q}^{WC}$, and the image observation $h = \begin{pmatrix}u & v\end{pmatrix}^\top$, which determine the depth priorρ_0, σ_0.

The end of the initialization ray is the current camera location estimate,

$$\begin{pmatrix}\hat{x}_i & \hat{y}_i & \hat{z}_i\end{pmatrix}^\top = \hat{\mathbf{r}}^{WC} \tag{1.23}$$

The direction of the ray is computed from the observed point in the world coordinate frame,

$$\mathbf{h}^W = \mathbf{R}_{WC}\left(\mathbf{q}^{\hat{W}C}\right)\begin{pmatrix}\upsilon & \nu & 1\end{pmatrix}^\top \tag{1.24}$$

Where υ and ν are normalized image coordinates. The angles we parametrize the direction in are

$$\begin{pmatrix}\theta_i \\ \phi_i\end{pmatrix} = \begin{pmatrix}\arctan\left(\mathbf{h}_x^W, \mathbf{h}_z^W\right) \\ \arctan\left(-\mathbf{h}_y^W, \sqrt{\mathbf{h}_x^{W\,2} + \mathbf{h}_z^{W\,2}}\right)\end{pmatrix}. \tag{1.25}$$

The covariances of $\hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{\theta}_i$, and $\hat{\phi}_i$ are derived from the image measurement error covariance $\mathbf{R}_i$ and the state covariance estimate $\hat{\mathbf{P}}_{k|k}^{\text{new}}$.

The state covariance after feature initialization is

$$\hat{\mathbf{P}}_{k|k}^{\text{new}} = \mathbf{J}\begin{pmatrix}\hat{\mathbf{P}}_{k|k} & 0 & 0 \\ 0 & \mathbf{R}_i & 0 \\ 0 & 0 & \sigma_\rho^2\end{pmatrix}\mathbf{J}^\top \tag{1.26}$$

$$\mathbf{J} = \left(\begin{array}{c|c} I & 0 \\ \hline \frac{\partial \mathbf{y}}{\partial \mathbf{r}^{WC}}, \frac{\partial \mathbf{y}}{\partial \mathbf{q}^{WC}}, 0, \ldots, 0, & \frac{\partial \mathbf{y}}{\partial \mathbf{h}}, \frac{\partial \mathbf{y}}{\partial \rho} \end{array} \right). \tag{1.27}$$

1.6 Switching from Inverse Depth to XYZ

Conversion from Inverse Depth to XYZ Coding

After each estimation step, the linearity index L_d is computed for every map feature coded in inverse depth

$$\mathbf{h}^W_{XYZ} = \hat{\mathbf{x}}_i - \hat{\mathbf{r}}^{WC} \qquad \sigma_d = \frac{\sigma_\rho}{\rho_i^2} \qquad \sigma_\rho = \sqrt{\mathbf{P}_{\mathbf{y}_i \mathbf{y}_i}(6,6)} \tag{1.28}$$

$$d_i = \left\| \mathbf{h}^W_{XYZ} \right\| \qquad \cos\alpha = \mathbf{m}^\top \mathbf{h}^W_{XYZ} \left\| \mathbf{h}^W_{XYZ} \right\|^{-1}. \tag{1.29}$$

where $\hat{\mathbf{x}}_i$ and $\mathbf{P}_{y_i y_i}$ is the submatrix 6x6 covariance matrix corresponding to the considered feature.

If L_d is below a certain switching threshold, the feature is transformed using Equation 1.5 and the full state covariance matrix P is transformed with the corresponding Jacobian,

$$\mathbf{P}_{\text{new}} = \mathbf{J}\mathbf{P}\mathbf{J}^\top \qquad \mathbf{J} = \text{diag}\left(\mathbf{I}, \frac{\partial \mathbf{x}_i}{\partial \mathbf{y}_i}, \mathbf{I}\right). \tag{1.30}$$

Linearity Index Threshold

They use $L_d < 0.1$ as a threshold to determine linearity and to switch to XYZ encoding.